

IN THE CLAIM

1       1. (Currently Amended) A method for augmenting a debugger having debugging  
2            functionality used to debug a first program, comprising the steps of:  
3                providing the debugger;  
4                inputting a debugging command at a debugging prompt provided by the  
5                debugger;  
6                providing a second program having second-program functionality; and  
7                providing integration code for invoking a piece of code to perform a  
8                task[[s]] in response to [[a]] the debugging command, based on  
9                types of a breakpoint;  
10            wherein if the breakpoint is a debugging breakpoint, then the piece of code  
11                is selected from the debugging functionality, else if the breakpoint  
12                is an instrumentation breakpoint, then the piece of code is selected  
13                from the second-program functionality.

14       2. (Original) The method of claim 1 further comprises the step of using an instrumentor as  
15            the second program.

1       3. (Canceled)

1       4. (Previously Presented) The method of claim 1 further comprises the step of making the  
2            piece of code an executable part of the first program.

1       5. (Previously Presented) The method of claim 1 further comprises the step of using a  
2            trampoline as the piece of code.

1 6. (Previously Presented) The method of claim 1 wherein the debugging command is  
2 selected from one or a combination of:  
3 input from a user using the debugger;  
4 a script file associated with the first program; and  
5 a configuration file associated with the first program.

1 7. (Canceled)

1 8. (Previously Presented) The method of claim 1 further comprises the steps of:  
2 integrating the debugger, the second program, and the integration code  
3 into a combined code; and  
4 embedding the combined code into a language environment.

1 9. (Previously Presented) The method of claim 8 further comprises the step of using an  
2 Integrated Development Environment as the language environment.

1 10. (Currently Amended) A system for augmenting a debugger having debugging  
2 functionality used to debug a first program, comprising:  
3 the debugger;  
4 a debugging command being inputted at a debugging prompt provided by  
5 the debugger;  
6 a second program having second-program functionality; and  
7 integration code for  
8 invoking a piece of code to perform a task in response to [[a]] the  
9 debugging command, based on types of a breakpoint;

10 wherein if the breakpoint is a debugging breakpoint, then the piece of code  
11 is selected from the debugging functionality, else if the breakpoint  
12 is an instrumentation breakpoint, then the piece of code is selected  
13 from the second-program functionality.

1 11. (Original) The system of claim 10 further comprises an instrumentor used as the  
2 second program.

1 12. (Canceled)

1 13. (Previously Presented) The system of claim 10 wherein the piece of code is an  
2 executable part of the first program.

1 14. (Previously Presented) The system of claim 10 wherein a trampoline is used as the  
2 first piece of code.

1 15. (Previously Presented) The system of claim 10 wherein the debugging command is  
2 selected from one or a combination of:  
3 input from a user using the debugger;  
4 a script file associated with the first program; and  
5 a configuration file associated with the first program.

1 16. (Canceled)

1 17. (Previously Presented) The system of claim 10 wherein:

2 the debugger, the second program, and the integration code are integrated  
3 into a combined code; and  
4 the combined code is embedded in a language environment.

1 18. (Previously Presented) The system of claim 17 wherein an Integrated Development  
2 Environment is used as the language environment.

1 19. (Currently Amended) A computer-readable medium embodying instructions that  
2 cause a computer to perform a method for augmenting a debugger having  
3 debugging functionality used to debug a first program, the method comprising the  
4 steps of:

5 providing the debugger;  
6 inputting a debugging command at a debugging prompt provided by the  
7 debugger;  
8 providing a second program having second-program functionality; and  
9 providing integration code for  
10 invoking a piece of code to perform a task in response to [[a]] the  
11 debugging command, based on types of a breakpoint;  
12 wherein if the breakpoint is a debugging breakpoint, then the piece of code  
13 is selected from the debugging functionality, else if the breakpoint  
14 is an instrumentation breakpoint, then the piece of code is selected  
15 from the second-program functionality.

1 20. (Original) The computer-readable medium of claim 19 wherein the method further  
2 comprises the step of using an instrumentor as the second program.

1 21. (Previously Presented) The method of claim 1 wherein the piece of code is stored in a  
2 library.

1 22. (Previously Presented) The method of claim 1 wherein the integration code generates  
2 the piece of code.

1 23. (Previously Presented) The method of claim 1 wherein the integration code keeps  
2 track of modifications to the first program, and, if appropriate, undoes the  
3 modifications.

1 24. (Previously Presented) The method of claim 1 wherein in a loop of more than one  
2 time, execution of code in the loop is transferred to the debugger one time.

1 25. (Previously Presented) The system of claim 10 wherein the piece of code is stored in a  
2 library.

1 26. (Previously Presented) The system of claim 10 wherein the integration code generates  
2 the piece of code.

1 27. (Previously Presented) The system of claim 10 wherein the integration code keeps  
2 track of modifications to the first program, and, if appropriate, undoes the  
3 modifications.

1 28. (Previously Presented) The system of claim 10 wherein in a loop of more than one  
2 time, execution of code in the loop is transferred to the debugger one time.